college_student_placement.

let's import useful libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

uploading dataset

In [83]:

```
students = pd.read_csv(r"C:\Users\taigk\OneDrive\Documents\Desktop\new_journey_78
```

let's walk through the first steps

In [3]:

```
students            # to get first look of my dataset
```

Out[3]:

	College_ID	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Internship_Exper
0	CLG0030	107	6.61	6.28	8	
1	CLG0061	97	5.52	5.37	8	
2	CLG0036	109	5.36	5.83	9	
3	CLG0055	122	5.47	5.75	6	
4	CLG0004	96	7.91	7.69	7	
...	...	...	...	...	...	
9995	CLG0021	119	8.41	8.29	4	
9996	CLG0098	70	9.25	9.34	7	
9997	CLG0066	89	6.08	6.25	3	
9998	CLG0045	107	8.77	8.92	3	
9999	CLG0060	109	9.41	9.77	8	

10000 rows × 10 columns

In [4]:

```
students.head()         #to get first five row
```

Out[4]:

	College_ID	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Internship_Experien
0	CLG0030	107	6.61	6.28	8	N
1	CLG0061	97	5.52	5.37	8	N
2	CLG0036	109	5.36	5.83	9	N
3	CLG0055	122	5.47	5.75	6	Y
4	CLG0004	96	7.91	7.69	7	N

In [5]:

```
students.tail()         #to get last five row
```

Out[5]:

	College_ID	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Internship_Exper
9995	CLG0021	119	8.41	8.29	4	
9996	CLG0098	70	9.25	9.34	7	
9997	CLG0066	89	6.08	6.25	3	
9998	CLG0045	107	8.77	8.92	3	
9999	CLG0060	109	9.41	9.77	8	

In [6]:

```
students.shape            #to get to know about the dataset rows and columns
```

Out[6]:

```
(10000, 10)
```

In [7]:

```
students.columns            #to get to know about the columns name
```

Out[7]:

```
Index(['College_ID', 'IQ', 'Prev_Sem_Result', 'CGPA', 'Academic_Performance',
       'Internship_Experience', 'Extra_Curricular_Score',
       'Communication_Skills', 'Projects_Completed', 'Placement'],
      dtype='object')
```

In [8]:

```
students.info()           #to get to know about the data types of my dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   College_ID              10000 non-null  object 
 1   IQ                      10000 non-null  int64  
 2   Prev_Sem_Result         10000 non-null  float64
 3   CGPA                    10000 non-null  float64
 4   Academic_Performance    10000 non-null  int64  
 5   Internship_Experience   10000 non-null  object 
 6   Extra_Curricular_Score  10000 non-null  int64  
 7   Communication_Skills    10000 non-null  int64  
 8   Projects_Completed      10000 non-null  int64  
 9   Placement               10000 non-null  object 
dtypes: float64(2), int64(5), object(3)
memory usage: 781.4+ KB
```

In [9]:

```
students.describe()             #to get done with thw arithmetic function
```

Out[9]:

	IQ	Prev_Sem_Result	CGPA	Academic_Performance	Extra_Curri
count	10000.000000	10000.000000	10000.000000	10000.000000	10
mean	99.471800	7.535673	7.532379	5.546400	
std	15.053101	1.447519	1.470141	2.873477	
min	41.000000	5.000000	4.540000	1.000000	
25%	89.000000	6.290000	6.290000	3.000000	
50%	99.000000	7.560000	7.550000	6.000000	
75%	110.000000	8.790000	8.770000	8.000000	
max	158.000000	10.000000	10.460000	10.000000	

let's starts dealing with the data

In [10]:

```
students.duplicated()           # to get to know is there any duplicates
```

Out[10]:

```
0       False
1       False
2       False
3       False
4       False
        ...  
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

In [11]:

```
students.isnull().sum()           # to get to know is there any null value
```

Out[11]:

```
College_ID              0
IQ                      0
Prev_Sem_Result         0
CGPA                    0
Academic_Performance    0
Internship_Experience   0
Extra_Curricular_Score  0
Communication_Skills    0
Projects_Completed      0
Placement               0
dtype: int64
```

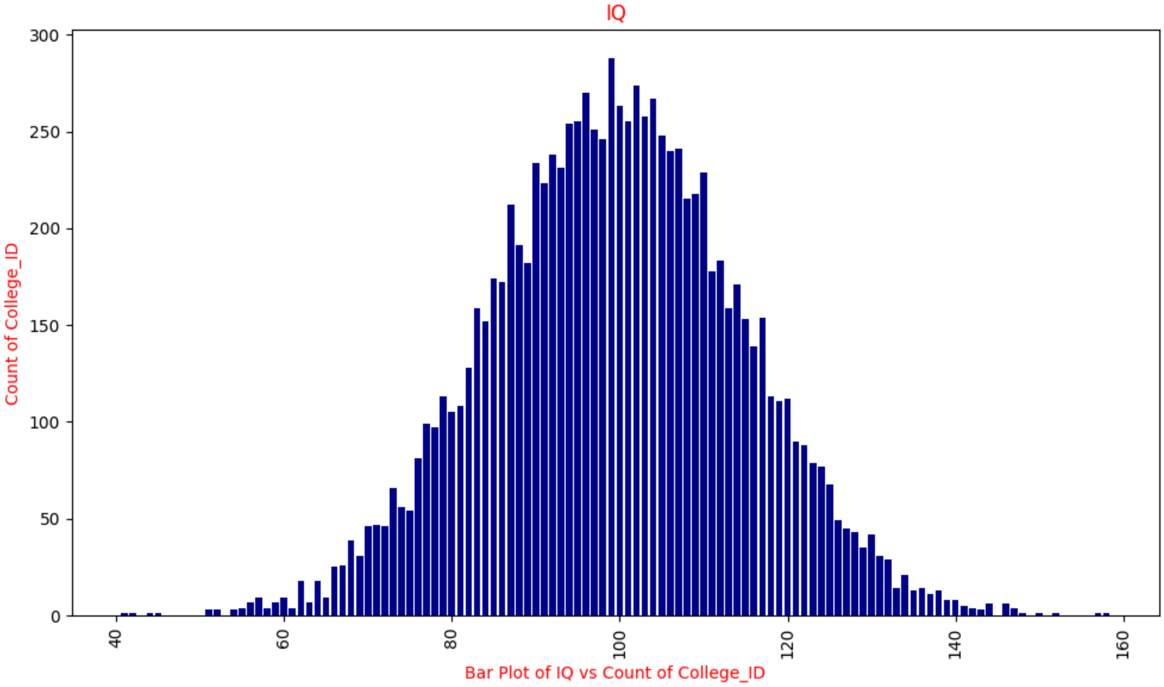
Let's walk through Bivariate Analysis

In [21]:

```
students_groupby = students.groupby('IQ')['College_ID'].count().reset_index()
```

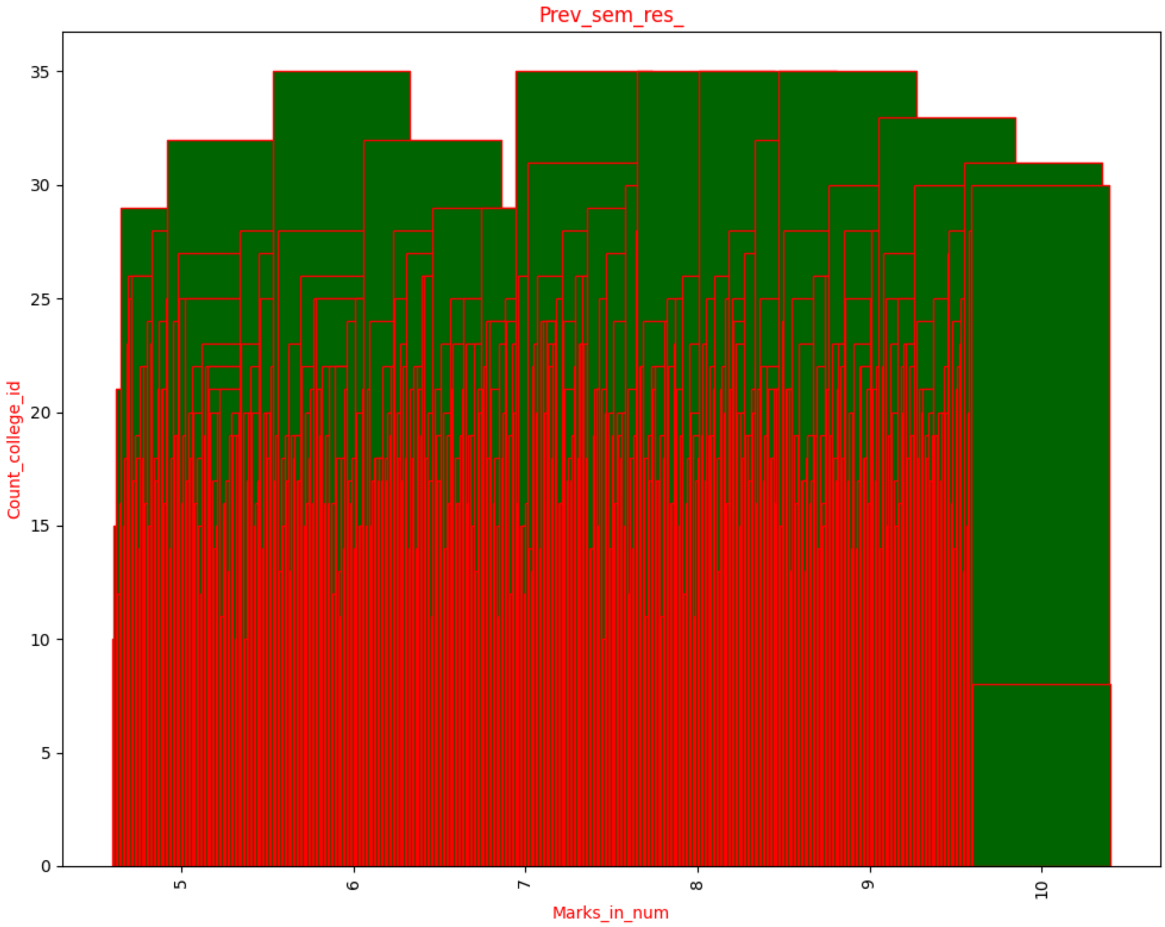
In [22]:

```
plt.figure(figsize= (10,6))           #Adjust the figure size
plt.bar(students_groupby['IQ'],students_groupby['College_ID'] , color='darkblue')
plt.title('IQ', c='r')
plt.ylabel('Count of College_ID', c='r')
plt.xlabel('Bar Plot of IQ vs Count of College_ID' ,c='r')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



In [31]:
```
student_prev=students.groupby('Prev_Sem_Result')['College_ID'].count().reset_index
```

In [39]:
```
plt.figure(figsize=(10,8) )  # adjust the figure size
plt.bar(student_prev['Prev_Sem_Result'], student_prev['College_ID'], color='darkgr
plt.title('Prev_sem_res_', c='r')
plt.ylabel('Count_college_id', c='r')
plt.xlabel('Marks_in_num', c='r')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

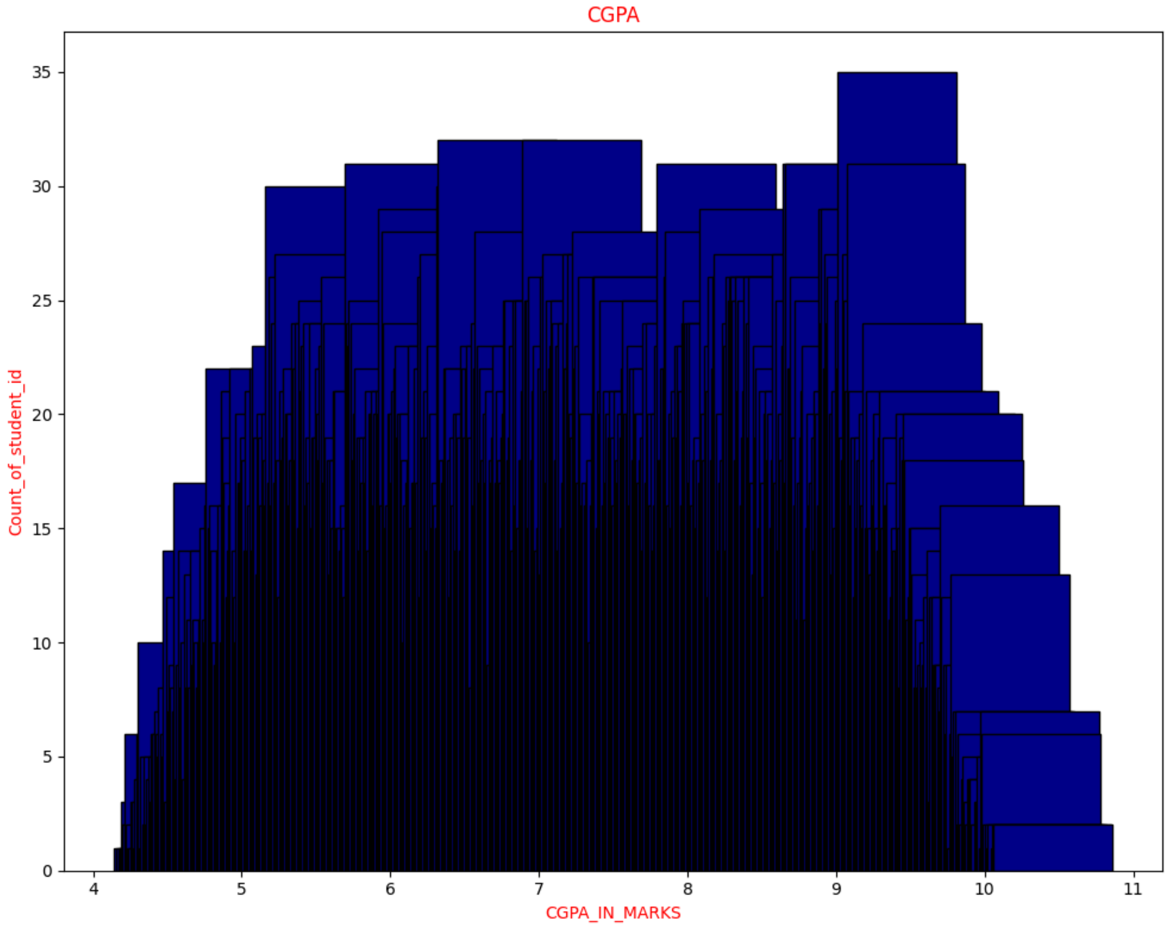


In [35]:

```
students_cg= students.groupby('CGPA')['College_ID'].count().reset_index()
```

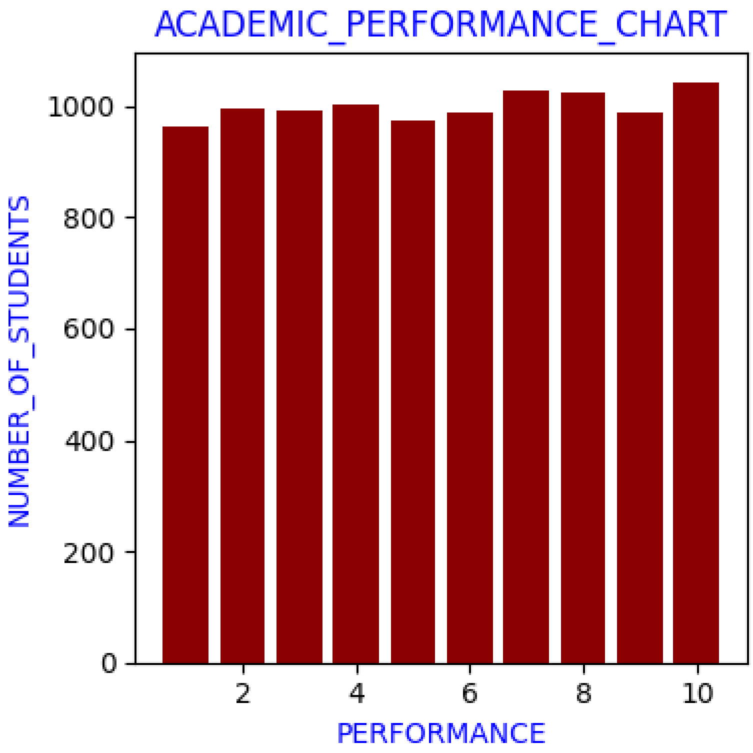
In [43]:

```
plt.figure(figsize=(10,8))
plt.bar(students_cg['CGPA'],students_cg['College_ID'] ,edgecolor='black', color='
plt.title('CGPA'  ,c='r')
plt.ylabel('Count_of_student_id' ,c='r')
plt.xlabel('CGPA_IN_MARKS', c='r')
plt.tight_layout()
plt.show()
```



In [47]:
```
students_per=students.groupby('Academic_Performance')['College_ID'].count().reset_
```

In [50]:
```
# graph that will show students performence
plt.figure(figsize=(4,4))
plt.bar(students_per['Academic_Performance'], students_per['College_ID'] , color=
plt.title('ACADEMIC_PERFORMANCE_CHART', c='b')
plt.ylabel('NUMBER_OF_STUDENTS' , c='b')
plt.xlabel('PERFORMANCE', c='b')
plt.tight_layout()
plt.show()
```



In [53]:

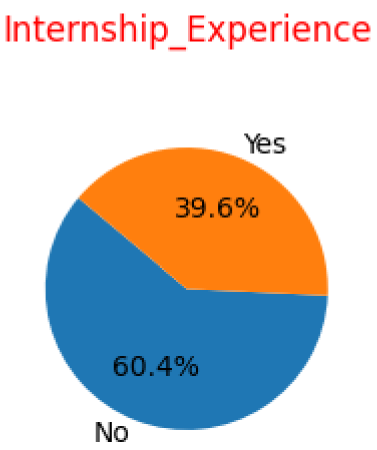
```
#students_int= students.groupby('Internship_Experience')['College_ID'].count().res
```

Out[53]:

	Internship_Experience	College_ID
0	No	6036
1	Yes	3964

In [61]:

```
#graph that will show the internship process
#grapgh while using DataFrame
student_int= pd.DataFrame({
    'Internship_Experience' : ['No', 'Yes'],
    'College_ID' : [6036, 3964]
})
plt.figure(figsize=(2,3))
values= student_int['College_ID'].values
labels = student_int['Internship_Experience']
plt.pie(values, labels=labels, autopct='%1.1f%%', startangle=140) , #color='Darkbl
plt.title('Internship_Experience', c='r')
plt.axis('equal') #that will ensure the pie is a circle
plt.show()
```

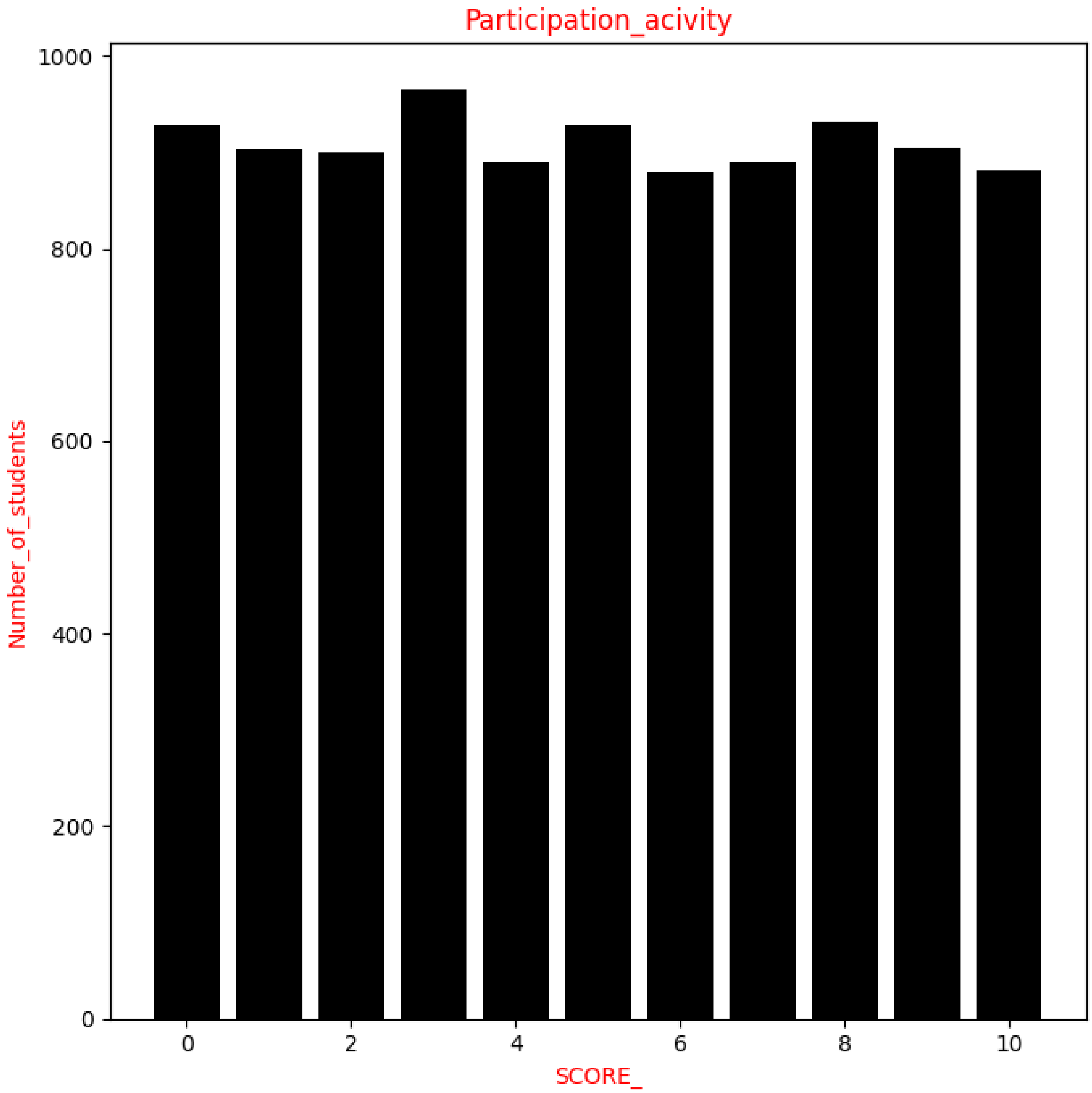


In [64]:

```
students_ext= students.groupby('Extra_Curricular_Score')['College_ID'].count().res
```

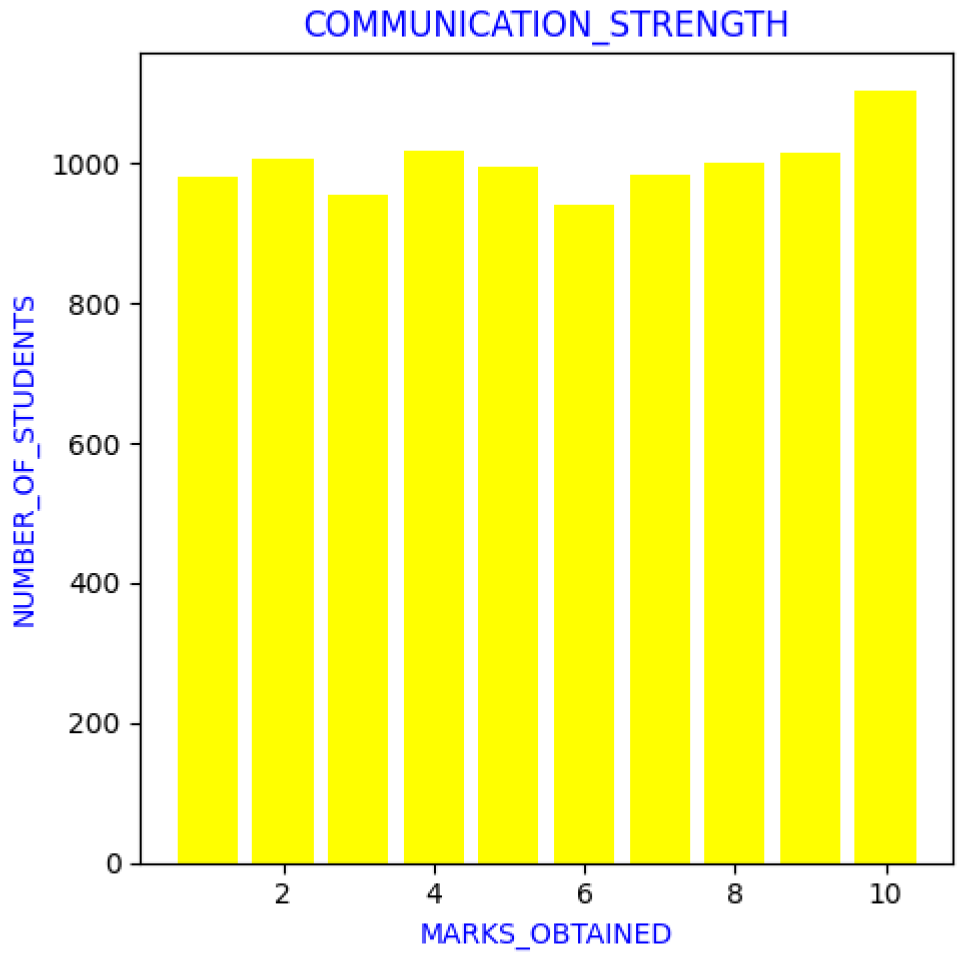
In [67]:

```
#that graph will show the extra_Score
plt.figure(figsize=(7, 7))
plt.bar(students_ext['Extra_Curricular_Score'], students_ext['College_ID'], color=
plt.title('Participation_acivity', c='r')
plt.ylabel('Number_of_students', c='r')
plt.xlabel('SCORE_', c='r')
plt.tight_layout()
plt.show()
```



```
In [70]: students_cum=students.groupby('Communication_Skills')['College_ID'].count().reset_
```

```
In [74]: #the graph that will show the students speaking score
         plt.figure(figsize=(5,5))
         plt.bar(students_cum['Communication_Skills'],students_cum['College_ID'] , color='y
         plt.title('COMMUNICATION_STRENGTH', c='b')
         plt.ylabel('NUMBER_OF_STUDENTS', c='b')
         plt.xlabel('MARKS_OBTAINED', c='b')
         plt.tight_layout()
         plt.show()
```



In [78]:

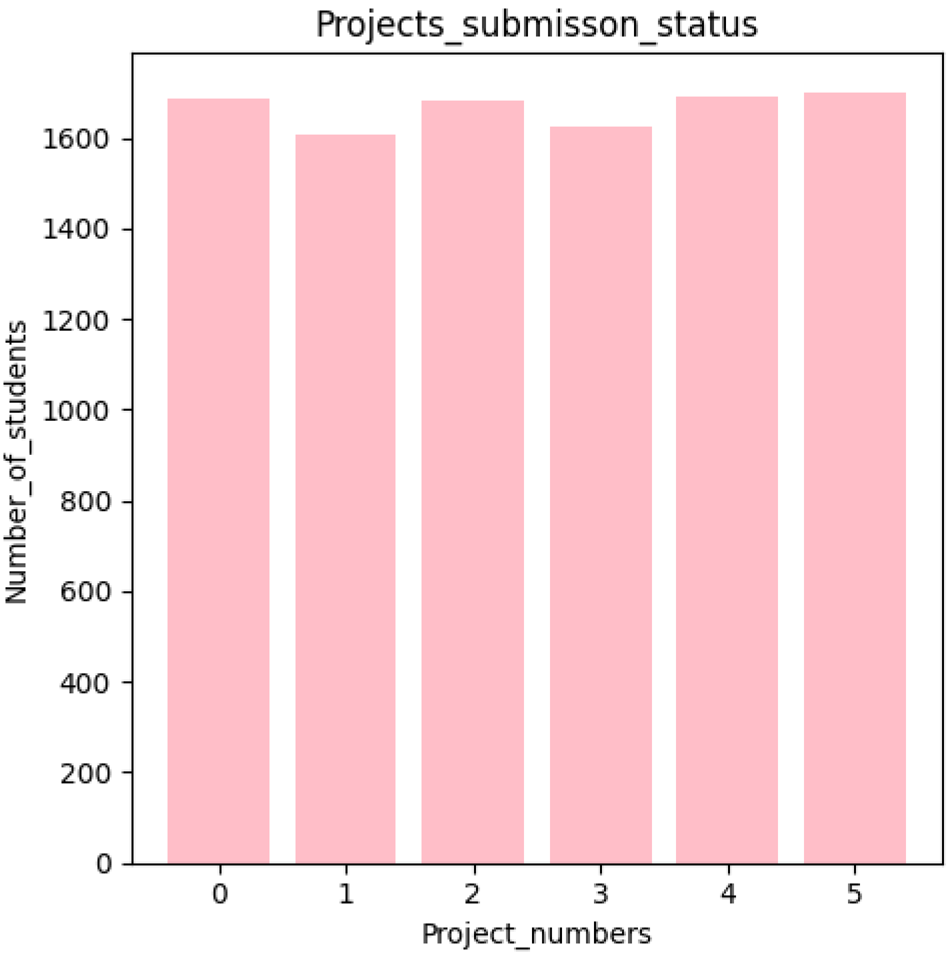
```
students_pro= students.groupby('Projects_Completed')['College_ID'].count().reset_i
```

Out[78]:

	Projects_Completed	College_ID
0	0	1688
1	1	1609
2	2	1681
3	3	1627
4	4	1693
5	5	1702

In [82]:

```
#the graph that will show project submission
plt.figure(figsize=(5,5))
plt.bar(students_pro['Projects_Completed'],students_pro['College_ID'], color='pink
plt.title('Projects_submisson_status', c='black')
plt.xlabel('Project_numbers' , c='black')
plt.ylabel('Number_of_students', c='black')
plt.tight_layout()
plt.show()
```



In [84]:

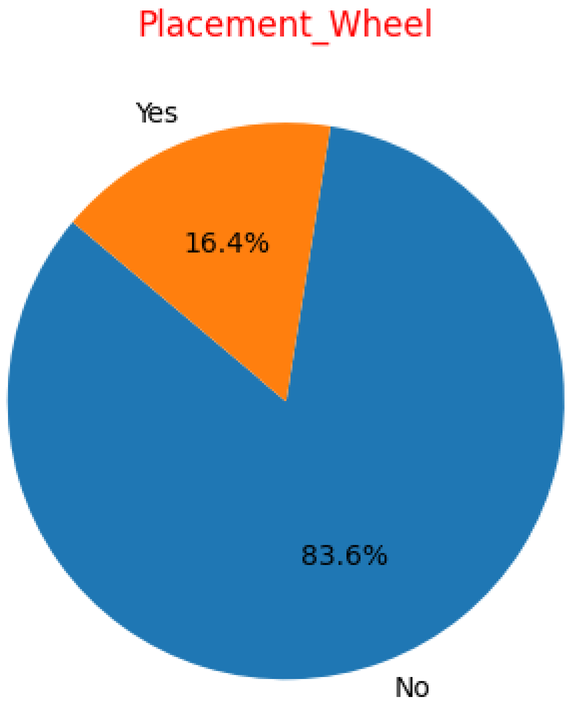
```
students.groupby('Placement')['College_ID'].count().reset_index()   #grouped by '
```

Out[84]:

	Placement	College_ID
0	No	8341
1	Yes	1659

In [91]:

```
#graph that will show the placements process/result
#graph while using DataFrame
students_place = pd.DataFrame({
    'Placement' : ['No', 'Yes'],
    'College_ID' : [8431,1659]
})
plt.figure(figsize=(4,4))
values = students_place['College_ID'].values
labels =students_place['Placement']
plt.pie(values, labels=labels, autopct='%1.1f%%', startangle = 140)
plt.title('Placement_Wheel', c='r')
plt.tight_layout()
plt.show()
```



In [ ]: